Customized Document Server (CDS)

By

Adnan Ali Khan

(2001-NUST-BIT-740)

A project report submitted for the fulfillment of the requirements for the degree of Bachelors in Information Technology

In

NUST Institute of Information Technology
National University of Sciences and Technology
Rawalpindi, Pakistan
(2005)

CUSTOMIZED DOCUMENT SERVER

(CDS)

By

Adnan Ali Khan

(2001-NUST-BIT-740)

A project report submitted in partial fulfillment of
the requirements for the degree of
Bachelors in Information Technology

NUST Institute of Information Technology
National University of Sciences & Technology
Rawalpindi, Pakistan
(2005)

CERTIFICATE

Certified that contents and form of the documentation entitled **Customized Document Server** submitted by Mr. Adnan Ali Khan has been found satisfactory for requirement of the Bachelors in Information Technology at NUST Institute of Information Technology.

Advisor: ____________________

Wg. Cdr. (R) Nasir Mahmood

Co-Advisor: ____________________

Dr. Sharifullah Khan

Committee Member: ____________________

Mr. Kamran Munir

Committee Member: ____________________

Air Cdr. (R) Mansoor Shaukat

DEDICATION

In the name of Allah, the Most Gracious, the Most Merciful

To my parents and my teachers.

ACKNOWLEDGEMENTS

I would like to thank my BIT Final Project Committee Members for their precious time, attention and contribution. I would specially like to thank my advisor, Wg. Cdr. (R) Nasir Mahmood, for his valuable guidance, advices, and cooperation. I would like to thank Dr. Sharifullah Khan, Mr. Kamran Munir, and Air Cdr. (R) Mansoor Shaukat for their assistance and interest in the project.

I would also like to thank my friends, Nabeel, Mubeen, Farhan, Kazim, and others, for their valuable support and assistance.

Finally, I would like to thank the faculty of NUST Institute of Information Technology (NIIT) who improved my project with all their comments and suggestions.

ABSTRACT

In today's world, the Internet is a vital technology used by all kinds of people, for all kinds of tasks. For scientists and researchers, it is a revolutionary transformation in the way they conducted their research work. They have found a common platform where they can disseminate their knowledge and research work to help their fellow researchers. On they other hand, they have also immensely benefited through the knowledge and research work of other researchers. Therefore, the Internet has created a win-win situation for the researchers.

The Internet itself is an enormously vast domain, and it is not very easy here to search the relevant documents. For this reason, a location has to be set up where all the relevant documents are placed. That location is the document server. Apart from being an important repository for the storage of documents, it provides other important facilities. These facilities include powerful document search, and submission. Another vital feature of the system is the conversion of format of a document to another compatible format. Agenda and bulletin viewing facilities are also provided in the system.

The system is *customized* because different user will have different kinds of access to the documents. For example, student will not be able to view a document in the category of *agenda point*, because it is only relevant to a faculty member. Hence the document server will provide a *customized* result to different users.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	Application Programming Interface
DB	Database
DBMS	Database Management System
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IO	Input/Output
IR	Information Retrieval
IT	Information Technology
RDBMS	Relational Database Management System
URL	Universal Resource Locator

Chapter 1

INTRODUCTION

This chapter provides an overview of the project in the form of an abstract, motivation, problem background, scope, domain, objectives and deliverables of the project and then finally discussing the procedure that we have adopted for the project.

1.1 INTRODUCTION

A document server is a dedicated server that facilitates users to easily browse and search documents stored in the database. It also enables the user to submit a document to be stored in the database. Other tasks might include converting a file from one format to another. The term *customized* refers to the fact that the server shall be exclusive to the requirements of the NUST Institute of Information Technology (NIIT). A *Document Server* is used for storing and sharing document information. Summing it up, the basic operation of a document server is to *store* and *search for* documents in numerous formats.

1.2 MOTIVATION

The motivation behind setting up a customized document server is to make it significantly simple and easy for users to search for documents from a database. The files can be in different formats. Users might also be able to submit documents to be stored in the database.

Currently, there is no such server at NIIT. As it is well known that NIIT collaborates with CERN (European Organization for Nuclear Research) quite frequently, it would be quite interesting to observe that CERN has it its own document server. CERN Document Server (CDS) provides the facility, for its users, to browse over 650,000 bibliographic records, including 320,000 complete-text documents. CDS provides the search and submission services for the documents. In conjunction to this, CDS also offers the format conversion, scanning and agenda planning services to the users.

Obtaining a rough idea from the CDS, a similar document server can be designed, particularly tailored to meet the ever-growing number of documents, especially research papers, at NIIT.

1.3 PROBLEM BACKGROUND

At a university that deals with wide-ranging research and study, it is essential for researchers to share their findings at a mutual platform. Without this platform, it is literally impossible for a researcher to find all of the useful documents and research paper easily. Either he/she will have to meet with all of his fellow researchers and investigate their research papers or will have to contact the authority where all of the research papers are available. In both cases, he/she will have to make a lot of effort by meeting a lot of people, and scanning though a number of documents/papers.

NIIT is an important institute of a very reputed university. Research related studies are an integral part of the academic achievements at NIIT. Many faculty members as well as students are involved in different research related projects.

Apart from research related document, other important documents are also required to readily available to the NIIT students, like lecture presentation, or course prerequisites and outlines. Therefore a repository is very much needed to store these documents in an organized and orderly method.

1.4 SCOPE OF THE PROJECT

The system must be able to store, browse, modify, submit and scan documents on the server. It should also be able to convert the format of a document to another compatible format type. The agenda of different faculty members as well as the news and bulletin regarding NIIT can also be viewed.

1.5 DOMAIN OF THE PROJECT

The domain of the project is databases. It involves database management, information retrieval, searching and indexing of files, etc.

1.6 OBJECTIVES OF THE PROJECT

The main objective of this project is to establish a common platform. At this platform, the following functionalities can be provided:

- storage of documents/papers,
- browsing of documents/papers,
- modification of documents/papers,
- submission of documents/papers,
- scanning the documents/papers,
- format conversion of documents/papers (within compatible format types),

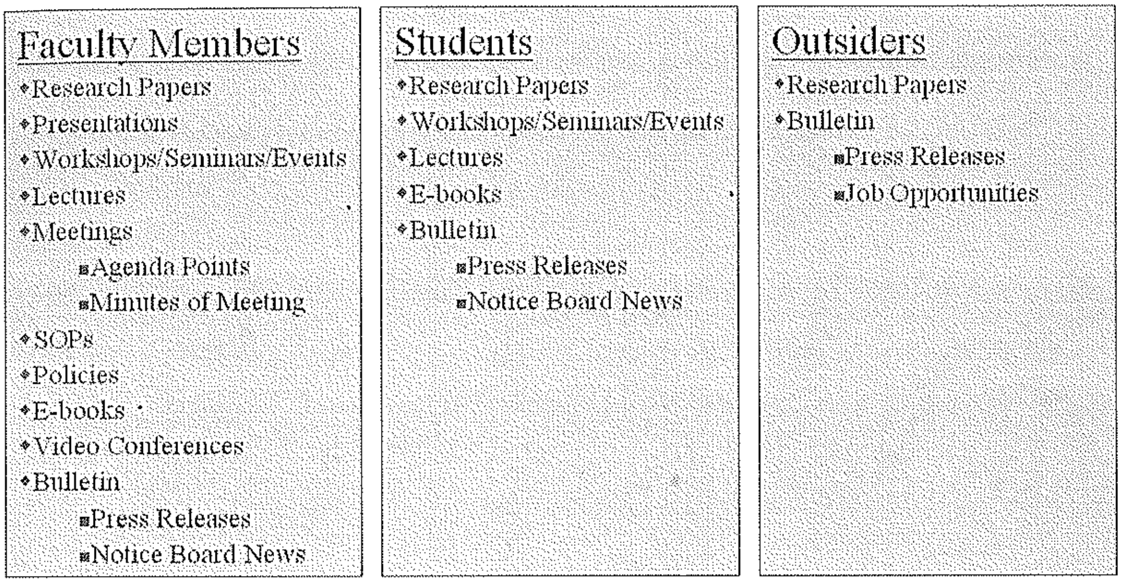
- finding document meetings, agenda points and presentations

1.7 TYPES OF DOCUMENTS

There are different types of documents that can be stored and accessed from the document server. These types are:

- Research Papers
- Presentations
- Workshops/Seminars/Events
- Lectures
- Meetings
    - Agenda Points
    - Minutes of Meeting
- SOPs (Standard Operation Procedures)
- Policies
- E-books
- Video Conferences
- Bulletin
    - Press Releases
    - Notice Board News

As different types of users would access the documents, they will be able to retrieve different types of documents. For example, a student does not need to retrieve a *minutes of meeting* document.



Faculty Members
- ♦Research Papers
- ♦Presentations
- ♦Workshops/Seminars/Events
- ♦Lectures
- ♦Meetings
  - ▪Agenda Points
  - ▪Minutes of Meeting
- ♦SOPs
- ♦Policies
- ♦E-books
- ♦Video Conferences
- ♦Bulletin
  - ▪Press Releases
  - ▪Notice Board News

Students
- ♦Research Papers
- ♦Workshops/Seminars/Events
- ♦Lectures
- ♦E-books
- ♦Bulletin
  - ▪Press Releases
  - ▪Notice Board News

Outsiders
- ♦Research Papers
- ♦Bulletin
  - ▪Press Releases
  - ▪Job Opportunities

Figure 1.1: Figure Access

1.8 ANTICIPATED USERS

The anticipate users of the server can be anyone interested in browsing through the stored documents. For privacy and security concerns, only authorized user might be allowed access to the documents.

For submitting the document, a user might either submit the documents himself/herself to the server, or forward it to some certified person, like administrator, who would submit it to the server on behalf of the actual user.

1.9 OUTLINE OF THE REPORT

This project report provides an insight into design and development of the Customized Document Server. In the following chapters, the background and the literature review of the project is explained. It is followed by the analysis and design of the project, then the project modules development and the system implementation is described. In the last chapter, the conclusion is drawn on the facts.

Chapter 2

LITERATURE REVIEW AND BACKGROUND STUDY

This chapter provides an overview of the background study of the project. It also provides explanations for some useful terms used throughout the project. Work related to the project is provided in the end.

2.1 BACKGROUND

With the 'computerization' of the world around us, every type of business and transaction has radically changed. Database Systems have replaced manual data recording and verifying methods. Relevant information is readily available even in the largest database systems.

Another revolutionary product of computer age is the Internet. A large amount of relevant data and information is accessible on just a single click. People in the Western countries even deal with matters regarding monetary transaction on the Internet.

The document server is an amalgamation of these two technologies, providing the client the access to the documents on the server-side machine through the Internet.

2.1.1 Document Server

A document server is a repository for (scientific) articles and papers. It provides powerful document searching facilities as well.

2.1.2 Web Server

A web server is a computer responsible for serving web pages, mostly HTML documents, via the HTTP protocol to clients, mostly web browsers.

Although web server programs differ in detail, they all share some basic common features. Every web server program operates by accepting HTTP requests from the network, and providing an HTTP response to the requester. The HTTP response typically consists of an HTML document, but can also be a raw text file, an image, or some other type of document.

2.1.3 Database

A database is a collection of data elements (facts) stored in a computer in a systematic way, such that a computer program can consult it to answer questions. The answers to those questions become information that can be used to make decisions that may not be made with the data elements alone.

At the core of the concept of a database is the idea of a collection of generic facts, or pieces of knowledge. Facts may be structured in a number of ways, known as database models. For instance, one database model is to associate each fact with a record representing an entity (such as a person), and to arrange these entities into trees or hierarchies -- the hierarchical database model. Another model is to arrange facts into sets of values which satisfy logical predicates -- the relational database model. Database management systems range from the extremely simple to the highly complex. Differences among DBMSes include whether they are capable of ensuring the integrity of the data; whether they may be used by many users at once; and what sorts of conclusions they can be programmed to compute from a set of data.

2.1.4 Database Implementation and Indexing

All kinds of database can take advantage of indexing to increase their speed. The most common kind of index is a sorted list of the contents of some particular table column, with pointers to the row associated with the value. An index allows a set of table rows matching some criterion to be located quickly. Various methods of indexing are commonly used; B-trees, hashes, and linked lists are all common indexing techniques.

Relational DBMSs have the advantage that indices can be created or dropped without changing existing applications, because applications don't use the indices directly. Instead, the database software decides on behalf of the application which indices to use. The database chooses between many different strategies based on which one it estimates will run the fastest.

Relational DBMSs utilize many different algorithms to compute the result of an SQL statement. The RDBMs will produce a plan of how to execute the query, which is generated by analysing the run times of the different algorithms and selecting the quickest. Some of the key algorithms that deal with joins are Nested Loops Join, Sort-Merge Join and Hash Join.

2.1.5 Database Management System

A database management system (DBMS) is a computer program (or more typically, a suite of them) designed to manage a database, a large set of

structured data, and run operations on the data requested by numerous users. Typical examples of DBMS use include accounting, human resources and customer support systems. Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

DBMS's are found at the heart of most database applications. Sometimes DBMSs are built around a private multitasking kernel with built-in networking support although nowadays these functions are left to the operating system.

2.1.6 Database Transaction

A database transaction is a unit of interaction with a database management system or similar system that is treated in a coherent and reliable way independent of other transactions. Ideally, a database system will guarantee all of the ACID properties for each transaction. In practice, these properties are often relaxed somewhat to provide better performance.

2.1.7 Session

A session is a connection between a client and a server.

2.1.8 Session Tracking

HTTP is 'stateless'. This means that between the time your browser receives a web page and asks for the next page, the server has forgotten who you are - in other words, when your browser asks for the second page, it has no way to know that it was the same browser that asked for the first page. This is obviously a problem for any application that needs to remember who you are - such as an application that requires a login. The notion of a single, unique user browsing from one page to another is referred to as a 'session'. As the web has evolved, several techniques for session tracking have evolved. The most common are cookies and URL-rewriting.

2.1.9 Information Retrieval

Information retrieval (IR) is the art and science of searching for information in documents, searching for documents themselves, searching for metadata which describes documents, or searching within databases, whether relational stand alone databases or hypertext networked databases such as the Internet or intranets, for text, sound, images or data. There is a common confusion, however, between data retrieval, document retrieval, information retrieval, and text retrieval, and each of these have their own bodies of literature, theory, praxis and technologies.

2.1.10 Index

An index can be defined on one or more columns in a table (these columns are referred to as the indexed columns). The index maps each set of indexed column values to the set of unique identifiers for the rows that have those column values. This structure provides a quick way to find the rows within a table using the indexed column(s).

2.1.11 Cookie

A cookie is a small amount (less than 1k usually) of text that a web server asks the web browser to store on the browser computer. This information is sent back to the server each time the browser makes a request for a URL on that server. This is the most common (and most preferred) method of session tracking.

2.1.12 HTTP

HTTP (Hypertext Transfer Protocol) is the protocol used between web browsers and web servers to transfer web pages and associated files (images, etc). It is the language of the World Wide Web. HTTP is built on top of TCP/IP.

HTTP is a request/response protocol between clients and servers. An HTTP client, such as a web browser, typically initiates a request by establishing a TCP connection to a particular port on a remote host (port 80 by default).

2.1.13 HTTP Transaction

HTTP transaction is a request sent from the browser to the server and the corresponding response from the server to the browser, both sent using HTTP. This round-trip communication path allows the browser to request a resource (URL) and receive a response from the server. It may include content sent by the browser (data entered in form fields, uploaded files) and content returned from the server (web page, image, etc).

2.1.14 Uniform Resource Locator

URL (Uniform Resource Locator) is a specially formatted string that describes a resource on the Internet. The browser uses this to determine where on the network the resource is located.

The URL syntax is designed to be generic, extensible, and able to express addresses in any character set using a limited subset of ASCII characters (for instance, whitespace is never used in a URL). URLs are classified by the "scheme" which typically identifies the network protocol used to retrieve the resource over a computer network.

2.1.15 Application Programming Interface

An application programming interface (API) is a set of definitions of the ways one piece of computer software communicates with another. It is a method of achieving abstraction, usually (but not necessarily) between lower-level and higher-level software.

One of the primary purposes of an API is to provide a set of commonly-used functions—for example, to draw windows or icons on the screen. Programmers can then take advantage of the API by making use of its functionality, saving them the task of programming everything from scratch. APIs are abstract: software that provides a certain API is often called the *implementation* of that API. In many instances, an API is synonymous with an SDK, or software development kit. An SDK may include an API as well as other tool/hardware, so the two terms are not strictly interchangeable.

2.1.16 Java Servlet API

The Java Servlet API allows a software developer to add *dynamic* content to a web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML. Servlets are the Java counterpart to dynamic web content technologies such as CGI or ASP. It has the ability to maintain state across many server transactions. This is done using HTTP Cookies, session variables of URL rewriting.

The Servlet API defines the expected interactions of a web container and a servlet. A web container is essentially the component of a web server that interacts with the servlets. The web container is responsible for mapping a URL to a particular servlet and ensuring that the URL requester has the correct access rights.

2.1.17 Servlet

A servlet is an object that receives requests and generates a response based on the request. The API defines HTTP subclasses of the generic servlet requests and responses as well as an HTTP session object that tracks multiple requests and responses between the web server and a client. Servlets may be packaged as a Web application.

2.1.18 Portable Document Format

Portable Document Format (PDF) is a file format developed by Adobe Systems for representing documents in a manner that is independent of the original application software, hardware, and operating system used to create those documents.

A PDF file can describe documents containing any combination of text, graphics, and images in a device independent and resolution independent format. These documents can be one page or thousands of pages, very simple or extremely complex with a rich use of fonts, graphics, colour, and images. PDF is an open standard, and anyone may write applications that can read or write PDFs royalty-free.

2.1.19 Apache HTTP Server

Apache HTTP Server is an open source HTTP web server for Unix-like systems (BSD, Linux, and UNIX systems), Microsoft Windows, Novell Netware and other platforms. Apache features highly configurable error messages, DBMS-based authentication databases, and content negotiation. It is also supported by several graphical user interfaces (GUIs) which permit easier, more intuitive configuration of the server. The Apache HTTP Server is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

Apache supports a variety of features, many implemented as compiled modules. These can range from server-side programming language support to authentication schemes.

2.2 RELATED WORK

CERN in Geneva, Switzerland is a world-renowned center famous for its research work. It deals with extensive research work relating to particle physics. To cope with the even increasing demand for the easy availability of relevant research papers and documents by other researchers, CERN set up an online document server. The CERN Document Server (CDS) has enlisted the availability of at least 360,000 research papers and documents. Other than from serving as a repository, CDS also provides the client the facility to submit their documents to be later used by other concerned individuals. Format conversion, agenda and bulletin viewing facilities are also supported by CDS.

Other than CDS, other research institutes also have their document servers, either online or not.

Adobe Document Server is also a good example of a well-made document server available online.

2.3 SUMMARY OF THE CHAPTER

This chapter provides an insight into the background study of the project. It also deals with the important terminologies used throughout the report, with their definitions. Previous work, related to the project, has also been specified in this chapter.

Chapter 3

METHODOLOGY

This chapter provides an insight to the methodologies and procedures used in order to accomplish the project. These procedures are provided in the form of work tasks, software process model used, programming languages, tools and hardware used.

3.1 WORK TASKS

The tasks for the completion of the project have been divided as follows:

- ❑ Requirements gathering
    - ○ Describe all the tasks that go into the instigation, scoping and definition of the system. Identifying the needs or requirements of a user to be in a position to design a solution.
- ❑ Evaluation of requirements and project plan
    - ○ After requirements gathering, the information about requirements was assessed to determine the relevance, effectiveness, and impact of the software. A project plan was shaped later on, based on the previous work.
- ❑ Database Design
    - ○ An effective, usable database is only possible if its designer has adequately incorporated the results of a full needs and task analysis. I understood the target user of the database, its intended functionality in

both the short and long term, and all system and human parameters affecting its use in order to design the database.

- □ Web pages Design
  - ○ For a frequent user, it must be easy to use the system effectively and efficiently. The interface or presentation of the system must be easy to understand and use. It must also be attractive. Keeping in mind these necessities, I design my web pages accordingly.
- □ Search Techniques
  - ○ Searching is the most important feature of the system. The searching has to be powerful as well as efficient. For this reason, I designed search algorithms, for searching records in the database, as well as through the documents.
- □ Database Search Techniques
  - ○ Searching records in the database, based on some user-specified parameters is also important. For that reason, I designed efficient search queries.
- □ Database and Web page implementation
  - ○ Finally, I implemented the theoretical homework into actual, reliable system modules. These were disintegrated part of the system.
- □ Integration

- The disintegrated modules of the system were integrated into a single system. An important factor for the integration was to make sure that these modules worked together agreeably.

- Testing
    - Testing was done consistently throughout the process. This was done to ensure the desired results are achieved without any consequences.

3.2 SOFTWARE PROCESS MODEL USED

The software process model that I have used for the execution of my project is the Waterfall Model. It is a conventional model compared to other software project models, like the Spiral Model or the Iterative Model. In this model, the development is seen as flowing steadily through the phases of requirements analysis, requirements specification, design, implementation, testing (validation), integration, and maintenance. This is a step-by-step method for implementing the software. I chose this to develop my software because its advantages include:

- Good progress tracking due to clear development stages.
- Milestones and deliverables can be clearly identified.
- Project Management and control is facilitated by the need to complete each stage before moving to the next.

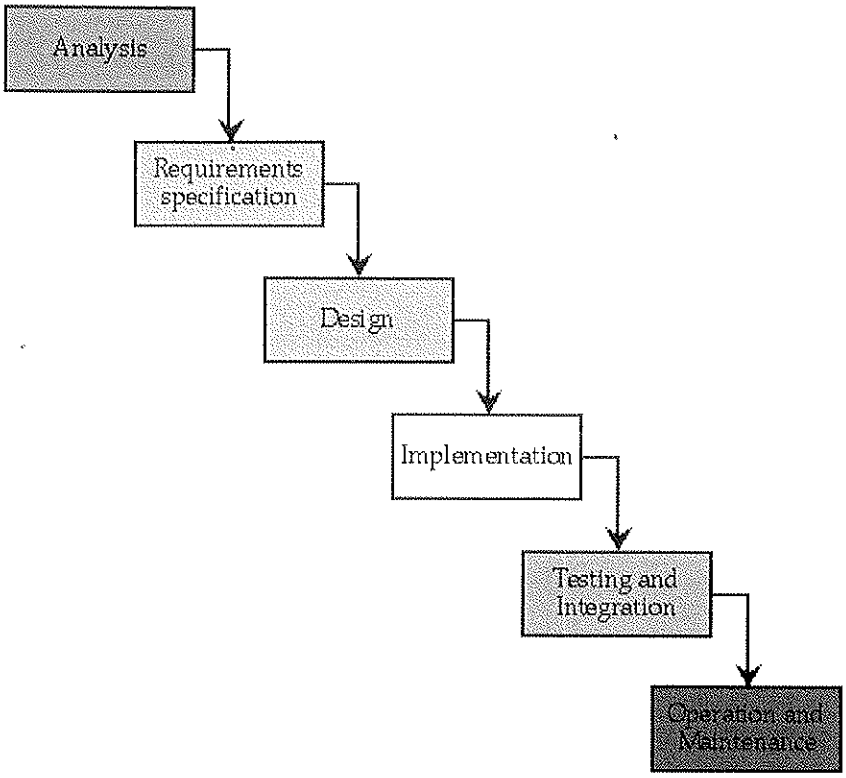


Figure 3.1: Waterfall Model

3.3 PROGRAMMING LANGUAGE USED

Java is used as the programming language with J2SDK1.4.2. The following Java APIs have been used extensively:

- ❑ Java Servlet API
- ❑ Java JDBC API
- ❑ Java Input/Output API
- ❑ Java Language and Utility API
- ❑ Java AWT API

- Google Desktop API
    - Version 0.9.4
- PDF Box API
    - Version 0.7.1

3.4 TOOLS USED

The following tools have been used extensively:

- JCreator
- Apache Tomcat Server
- SQL Server Enterprise Manager

3.5 HARDWARE USED

The development of the project has been done on a NIIT Lab Computer. Following are the hardware details:

- Intel Pentium-III
- 800 MHz Processor
- 256 MB RAM

3.6 SUMMARY OF THE CHAPTER

This chapter provides an insight into the methodology and approach the project has been developed and implemented. It includes the work tasks; the software process model for the project, the programming languages and the tools used as well as the hardware used.

Chapter 4

ANALYSIS AND DESIGN

This chapter provides an analysis of the project by discussing the different functionalities and utilities of the system. The design and plan of implementation is also provided in the end.

4.1 ANALYSIS

In the existing systems, like CDS, though documents are accessible on the Internet, but they treat every client in the same way. In CERN, no signing-up or signing-in is required. It means that whether you are a teacher or a student, a CERN member or an outsider, you can access the same documents. There is no customization in the list of search results to the client query. This may lead to the availability of important documents, even to the least related member.

We put the user customization facility in the system to take a step further in tackling different types of users.

4.1.1 Product Features

A few important deliverables of the customized document server are enlisted as follows:

- A powerful search engine,
- Document server processing entity,
- Website for the users to access the database,
- Server-side databases,

- ❑ Secure transaction of information

4.1.2 User Characteristics

The users of the software would generally be NIIT faculty members, students, and training branch members. Each user would have his own account to access the document server. Even if a NIIT faculty member wants retrieve to the documents, but he does not have an account, then he cannot access anything. The user, from any age group, can use the system as far as he has adequate knowledge of using different software. No special experience or technical expertise is required for the user to use the system.

4.1.3 Functional Requirements

A *functional requirement* is a requirement that specifies a mandatory behavior of the thing being specified. The functional requirements of the system are:

- ❑ Web browser
- ❑ Powerful search engine
- ❑ Saving of documents/papers on the databases at the server-side
- ❑ Accessibility of documents/papers
- ❑ Submitting documents/papers
- ❑ Converting documents from one format to another

4.1.4 Quality Attributes (Non-functional Requirements)

A *non-functional requirement* is a statement of how a system must behave; it is a constraint upon the systems behavior. The quality attributes (non-functional requirements) of the system are:

- □ Availability
- □ Scalability
- □ Performance
- □ Reliability
- □ Maintainability

4.1.5 Behavioral Requirements

Behavioral requirements define precisely what inputs the software expects, what outputs will be generated by the software, and the details of the relationships that exist between those inputs and outputs.

The system expects the inputs in the form of a word string, which specifies either the title of the documents/paper or the name of the author. Other searching factors can be the report number or the year that paper was published. As a result of the expected input, the system would generate a list of documents/papers, compatible with the search criteria. The format of the documents listed would be:

Title of the Document/ *Name of the Author(s)*

Report Number – Year Published

When a member wants to submit his document/paper, he would be given a list of categories to select the type of his document. For example, whether the document is from NIIT or from some external institute, or whether it is an internal note or a media file. Afterwards, the member would have to fill an information form, which will get the data, like name, date of birth, academic record, etc. When the form would be filled with appropriate data, the document will be sent to the administrative authority, which after analyzing the document would either accept or reject the submission. In either case, the member will be notified through an email. If his document is accepted, he would be assigned a report number.

To convert a document from one format to another, the user would be given a list of formats that have to be converted, and a list of formats the document has to be converted to. Once the compatible formats are chosen, the output of the system would be the document in new format. In the document server, different types of files are stored, from word documents to pictures. By compatible formats, it is meant any two formats that can be converted to each other. For example, from Microsoft Word (*.doc*) to Portable Document Format (*.pdf*). However, an image in JPEG format cannot be converted to a document in DOC format.

The user of the system can also view the agendas and time-tables of different committees or departments.

4.1.6 Searching for Documents

Document searching is the most important component of a document server. It makes it very easy for the client/user to exactly pinpoint to the required document once it is found. But searching documents also involves two types of methods:

- □ Database-documents searching
- □ Thorough-documents searching

In the database-documents searching, the system searches for the record entries in the database. This search can be based on a particular field, for example all the papers published in the year 2004, or all the documents written by Dr. Noam Chomsky. Once the search entry matches the database records, a list is generated regarding the particulars of these documents.

In the thorough-documents searching, the system searches 'through' the documents. What this means is that if an entry search word is written in a research paper or a document, it will be matched and displayed in a list regarding the particulars of these documents. For more than one entry search words, search by logical operators *AND* and *OR* is also available. In the 'AND' search, the documents containing only the search words are returned. In the 'OR' search, the documents containing either of the search words or both the search words are returned.

4.1.7 Submitting Documents

The client/user might want his/her document to be available to other concerned and related individuals, so that it might help in solving their queries and problems. For this reason, he might want to submit it on the document server, where most people turn to when they need help.

The client has the full liberty to submit his document in the document. But before the actual submission, he/she would have to provide some important information about them, like their name or telephone number. If he has submitted other documents before, he would not even be required to fill the information form again.

When the client/user submits the document, his information, and the document information is stored in the database, and the document is also stored in its specified location.

4.1.8 Format Conversion of the Documents

Since the client/user might want the documents in a particular format, for example, in Microsoft Word or PDF format, the document server provides the user the facility to convert his document to a specific format. Since two of the most widely used formats for research papers and documents are Microsoft Word and PDF, the system provides format conversion between these two.

After the format of the document has been converted, the client/user can easily download it to his computer by clicking on a link.

4.1.9 Viewing Agenda

In a university, students frequently need the assistance of different lecturers and faculty members. The students might have to visit their offices several times to meet them. They might not be available, delivering lecture somewhere or conducting lab sessions.

Therefore, to save precious time, the students might want to view the agendas of the faculty members. It can be made incredibly easy by posting their agendas on the document server. The students can view the agenda of the faculty member they want to visit, and meet them accordingly, hence saving a considerable amount of time and energy.

4.1.10 Viewing Bulletin

In a renowned university, different workshops, seminars and other events are frequently organized. The document server tackles the issue of posting this important information, so that users can view it on time and do not miss these events.

Other than this, additional information, like press releases, job opportunities, and notice board is also put up on the document server.

4.1.11 Session Tracking

When a client/user is signed-in, important information about him, like his identity, or duration of access time can be extracted. The user session starts when the client signs-in. It continues until the client/user signs-out from the document server.

4.2 DESIGN

4.2.1 Block Diagram

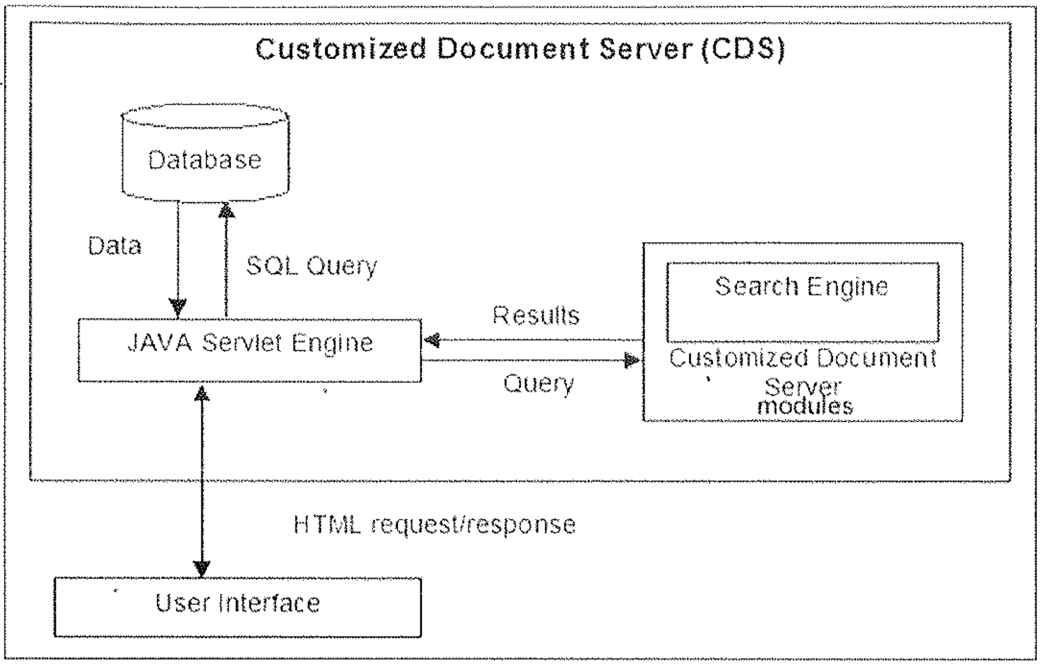


Figure 4.1: Block Diagram

The above architecture is made up of three major modules:

- **Search Engine**: The Search Engine module is used for searching through the document. The user specifies a word or a set of words, which are searched in the documents in the repository by the search engine. The result of the search by the engine is sent to the user through the Java Servlet Engine.

- **Java Servlet Engine**: The Java Servlet Engine module is used to get the HTML.request from the user and after the intended operation, respond to the user appropriately. This module executes java servlets and responds back to the user in HTML. This module creates dynamic web pages for the user by interacting with the search engine or the database. This module caters for storing and retrieving data from the database.

□ **Database:** The database module is a simple database, used to store, modify, and retrieve data. The request and results to the database are given and taken through the Java Servlet Engine module.

4.2.2 Entity-Relationship Diagram

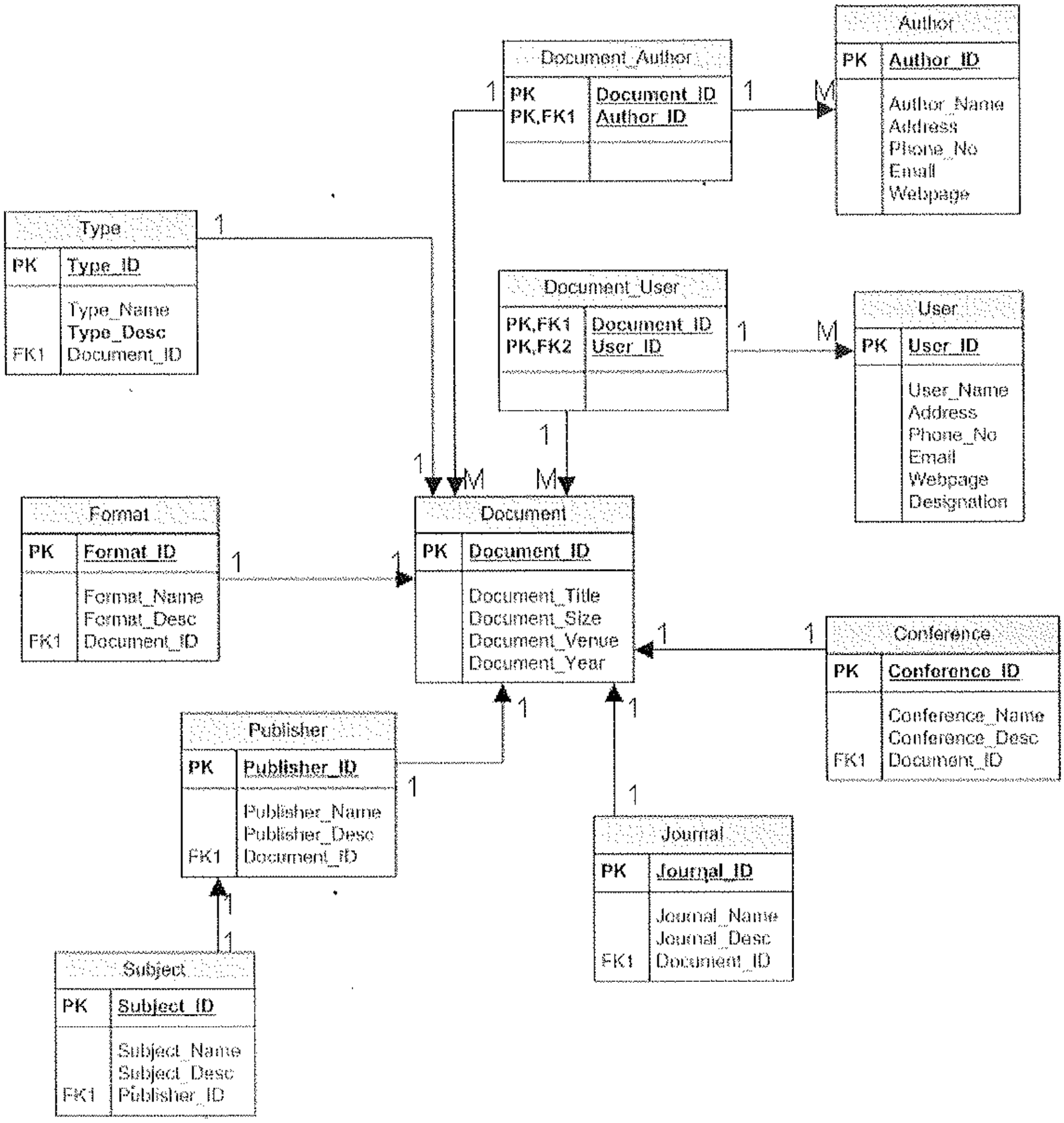


Figure 4.2: Entity-Relationship Diagram

4.2.3 Context Data Flow Diagram (DFD)

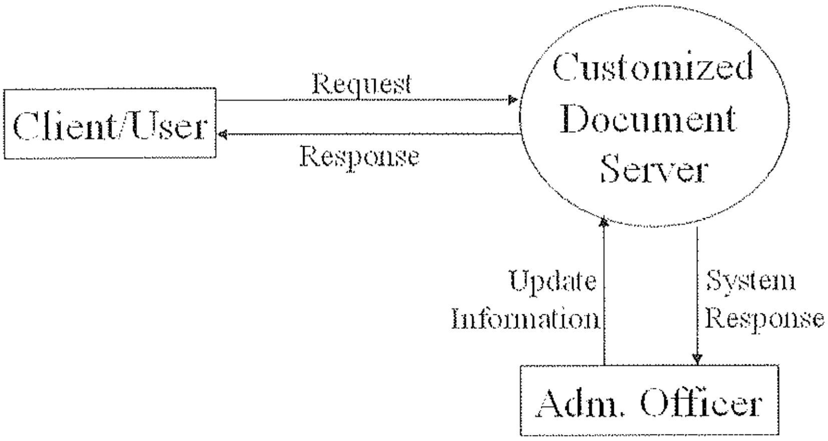


Figure 4.3: Context Data Flow Diagram

In the above diagram, the user request information. The big bubble signifies the system itself. Thus, in the simple diagram illustrated above, the user requests for relevant data from the document server system, in response to which, the system returns the intended data.

4.2.4 External Interface

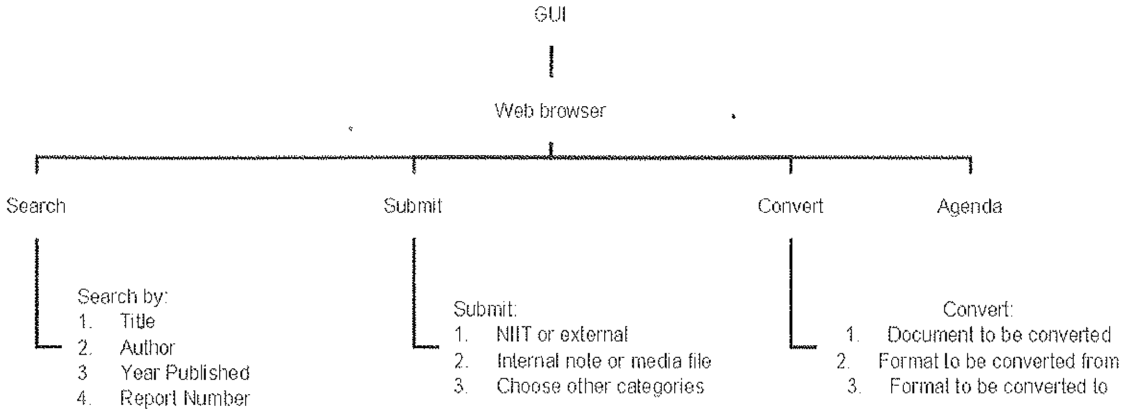

GUI
Web browser
Search
Submit
Convert
Agenda
Search by:
1. Title
2. Author
3 Year Published
4. Report Number
Submit:
1. NIIT or external
2. Internal note or media file
3. Choose other categories
Convert:
1. Document to be converted
2. Format to be converted from
3. Format to be converted to

Figure 4.4: External Interfaces

4.3 SUMMARY OF THE CHAPTER

This chapter provides an insight into the general analysis and design of the project. An in-depth analysis of the different features and functionalities of the project is provided. Afterwards, the design of the project is provided.

Chapter 5

IMPLEMENTATION AND VALIDATION

This chapter presents an outline of the techniques by which the system has been implemented. It also provides different snapshots and pictures to validate that the system works.

5.1 SYSTEM IMPLEMENTATION

The system is implemented as the approach discussed in the last chapter. Design for each component of the system is created and the components are integrated into the system afterwards. Following are the important system components:

- ❑ Search Engine
- ❑ Database
- ❑ Java Servlet Engine
- ❑ Session Tracking
- ❑ Format Conversion

5.1.1 Search Engine

The Search Engine module of the system deals with searching in through the database records and the documents. This has to be the most powerful component of the system as the major functionality of a document server is its ability to speedily search through the database and documents, and return the result.

The search entry page as well as the search response list page is developed using the Java Servlet API to generate the web pages dynamically. Java JDBC API is used for search through the database. Google Desktop API is used th search words through the documents.

5.1.2 Database

The database module of the system is used to store data, and later to retrieve it. It has to take care of the define constraints by the system manager and generate error if the data request is invalid.

The database is accessed through the Java Servlet Engine. The record search entry page as well as the search response page is developed using the Java Servlet API to generate the web pages dynamically. The database itself is made up in *SQL Server*.

5.1.3 Java Servlet Engine

The Java Servlet Engine module is the main module that helps in the interaction between the user and the system. It takes request from user from HTML web pages and also returns the results in form of the HTML web pages.

Apache HTTP Server is used as the web server. The Java Servlet API is used to dynamically generate web pages.

5.1.4 Session Tracking

The Session Tracking module is an inportant module that helps in maintaining the session of a user once he/she is signed-in. Important information about the user can be determined using the session tracking facility.

The user session starts when he/she signs-in. The session tracking facility is implemented using the Java Servlet API.

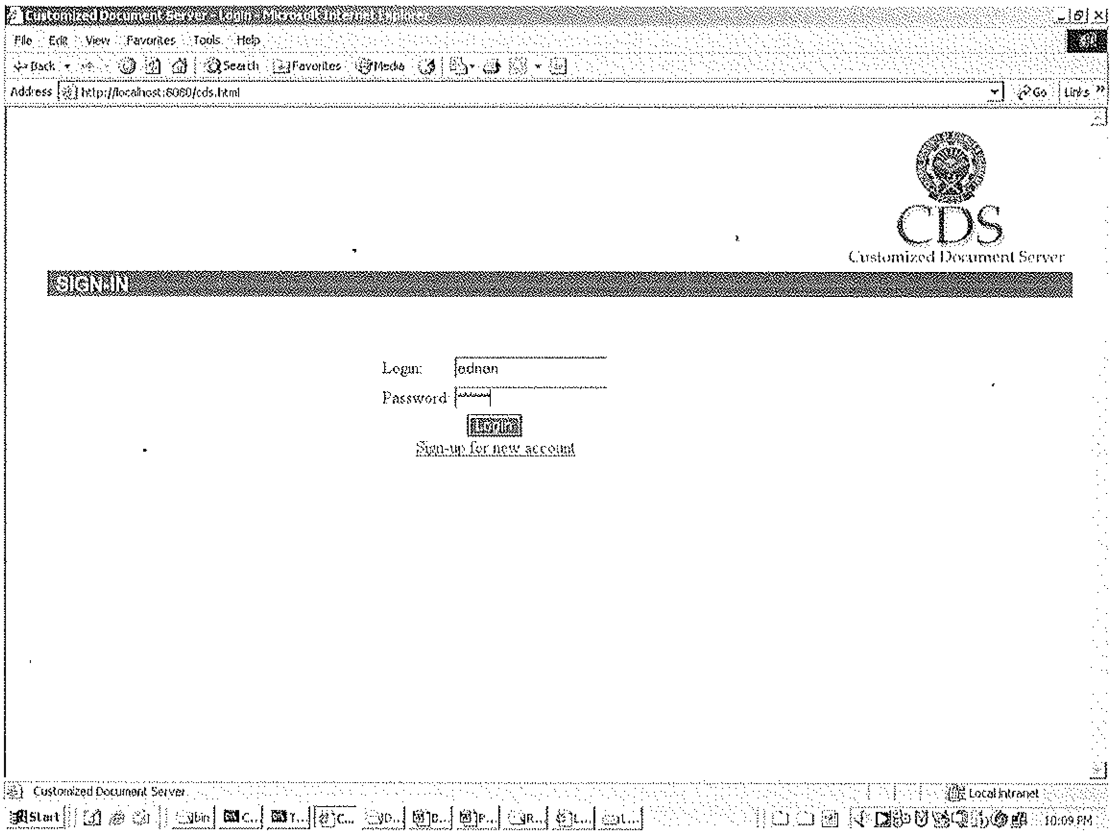
5.1.5 Format Conversion

The format conversion module helps the user to cinver the format of any documents he wants. The user sepcifies the location of the document that he wants to convert and the desired format, to which he wants to convert his document. Later her can download the 'converted' document through a link provided on the web page.

The format conversion module is implemented using the PDFBOX API. The requesting and responsing web pages for format conversion is implemented using the Java Servlet API.

5.2 SYSTEM VALIDATION

First of all, the user signs-in.



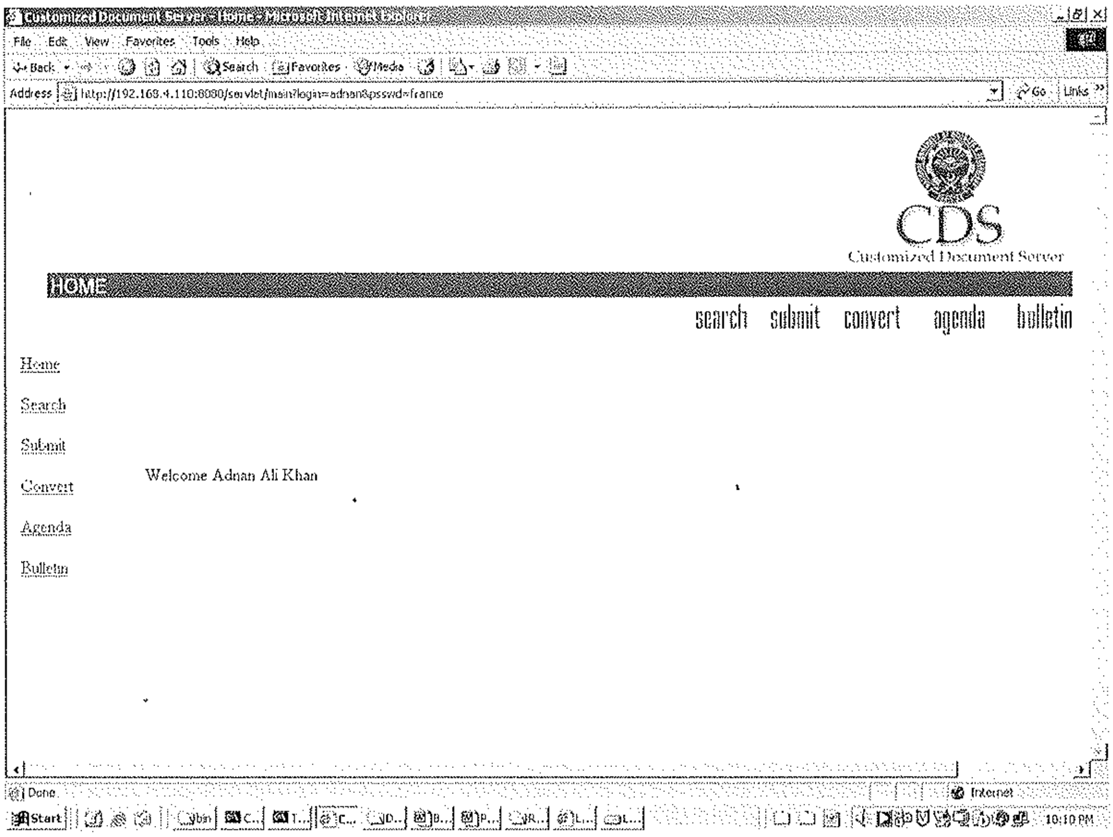
Customized Document Server - Login - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost:8080/cds.html
Go Links
CDS
Customized Document Server
SIGN-IN
Login
Sign-up for new account
Customized Document Server
Local intranet
Start
10:09 PM

Login: adnan

Password: ******

Figure 5.1: System Validation (a)

He is take to his homepage.



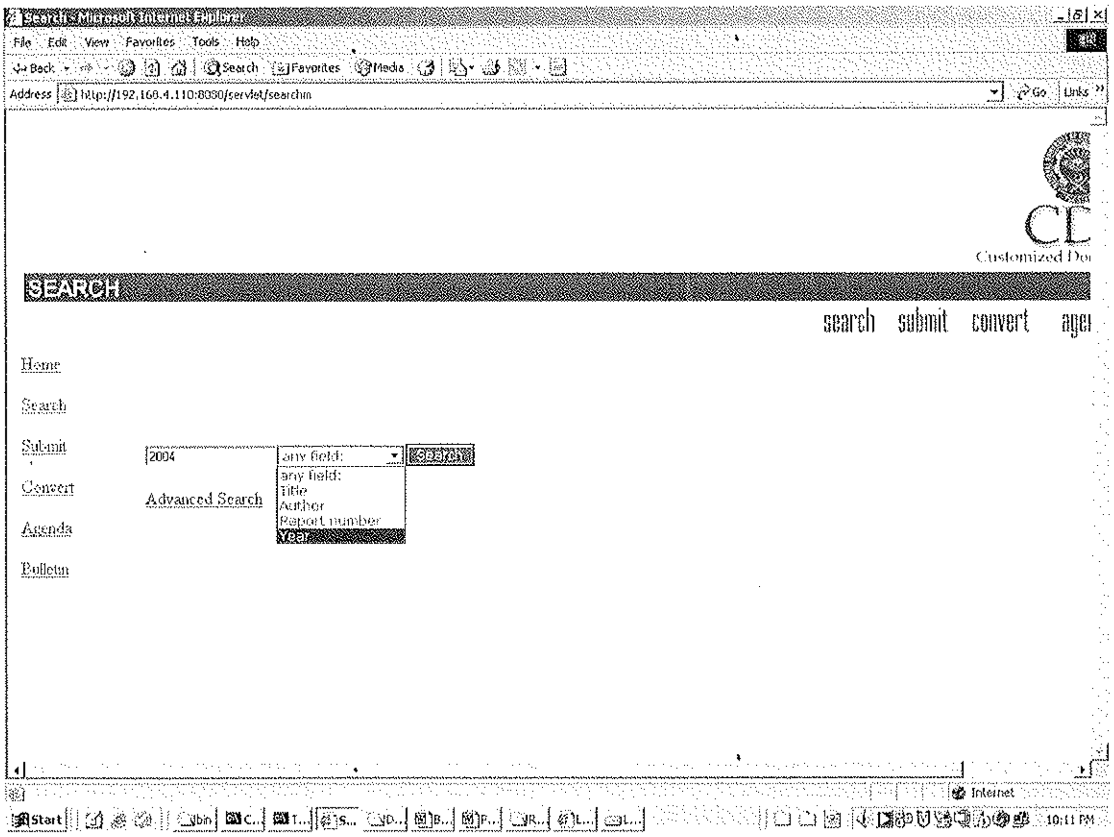
Customized Document Server - Home - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go Links
CDS
Customized Document Server
HOME
search submit convert agenda bulletin
Home
Search
Submit
Convert
Agenda
Bulletin
Done
Internet
Start
10:10 PM

Address: http://192.168.4.110:8080/servlet/main?login=adnan&psswd=france

Welcome Adnan Ali Khan

Figure 5.2: System Validation (b)

For database search, he can choose any field for priority:

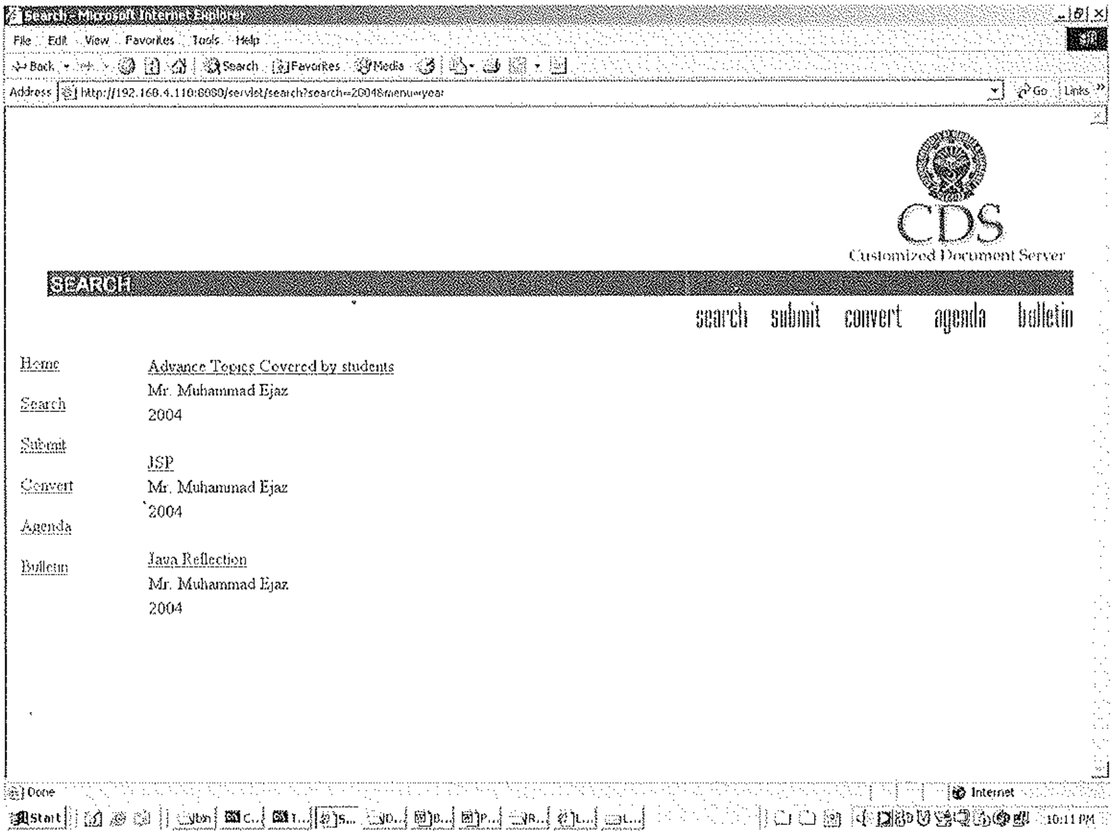


Address http://192.168.4.110:8080/servlet/searchm

2004

any field:

Figure 5.3: System Validation (c)



Search - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://192.168.4.110:8080/servlet/search?search=2004&menu=year
Go Links
CDS
Customized Document Server
SEARCH
search submit convert agenda bulletin
Home
Search
Submit
Convert
Agenda
Bulletin
Done
Internet
Start
10:11 PM

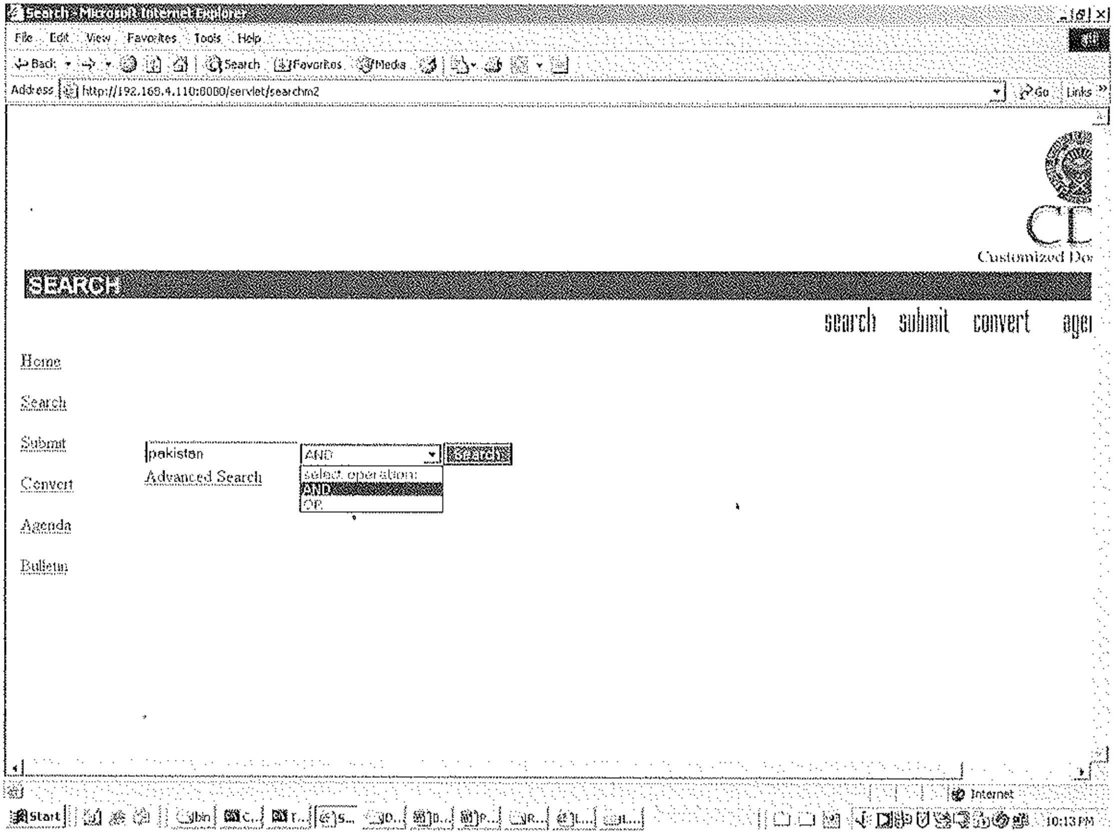
Advance Topics Covered by students
Mr. Muhammad Ejaz
2004

JSP
Mr. Muhammad Ejaz
2004

Java Reflection
Mr. Muhammad Ejaz
2004

Figure 5.4: System Validation (d)

For advanced search, choose the logical operation for the word:



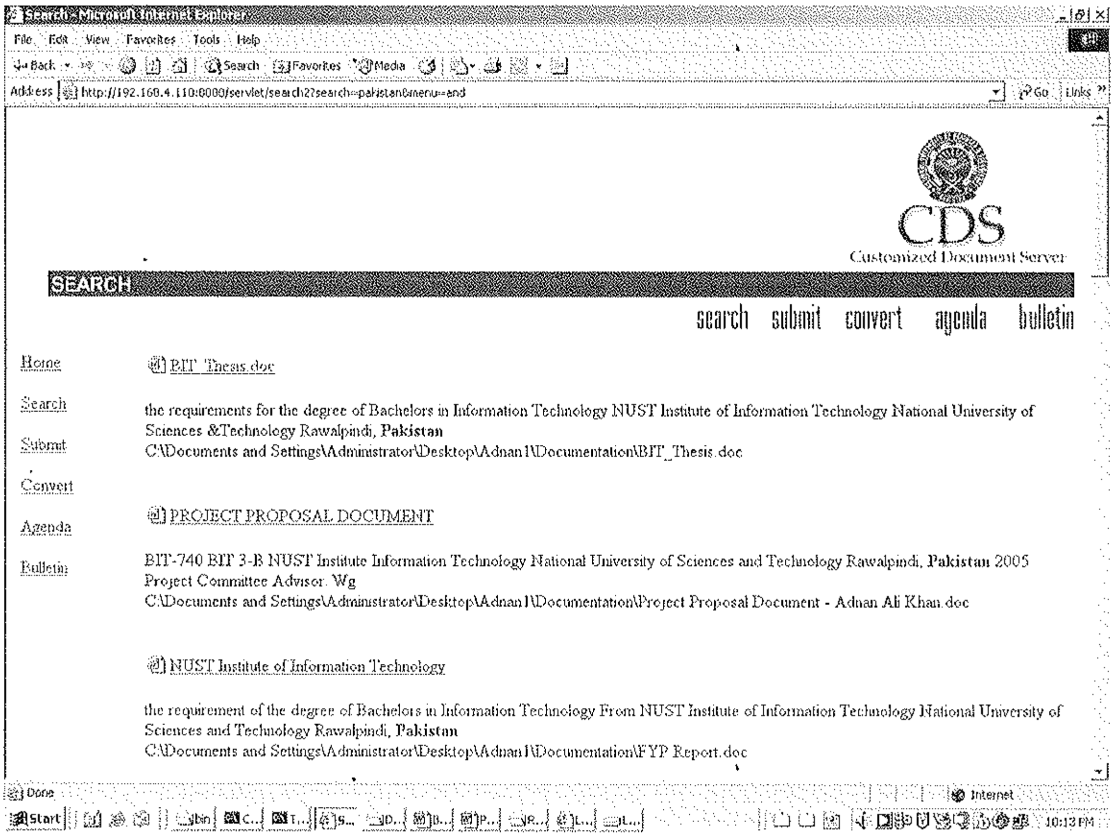
Address: http://192.168.4.110:8080/servlet/searchm2

pakistan

AND

Figure 5.5: System Validation (e)

The result of advanced search:


Search - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Go Links
CDS
Customized Document Server
SEARCH
search submit convert agenda bulletin
Home
Search
Submit
Convert
Agenda
Bulletin
Done
Internet
Start
10:13 PM

Address http://192.160.4.110:8000/servlet/search2?search=pakistan&menu=and

BIT_Thesis.doc

the requirements for the degree of Bachelors in Information Technology NUST Institute of Information Technology National University of Sciences &Technology Rawalpindi, **Pakistan**
C:\Documents and Settings\Administrator\Desktop\Adnan1\Documentation\BIT_Thesis.doc

PROJECT PROPOSAL DOCUMENT

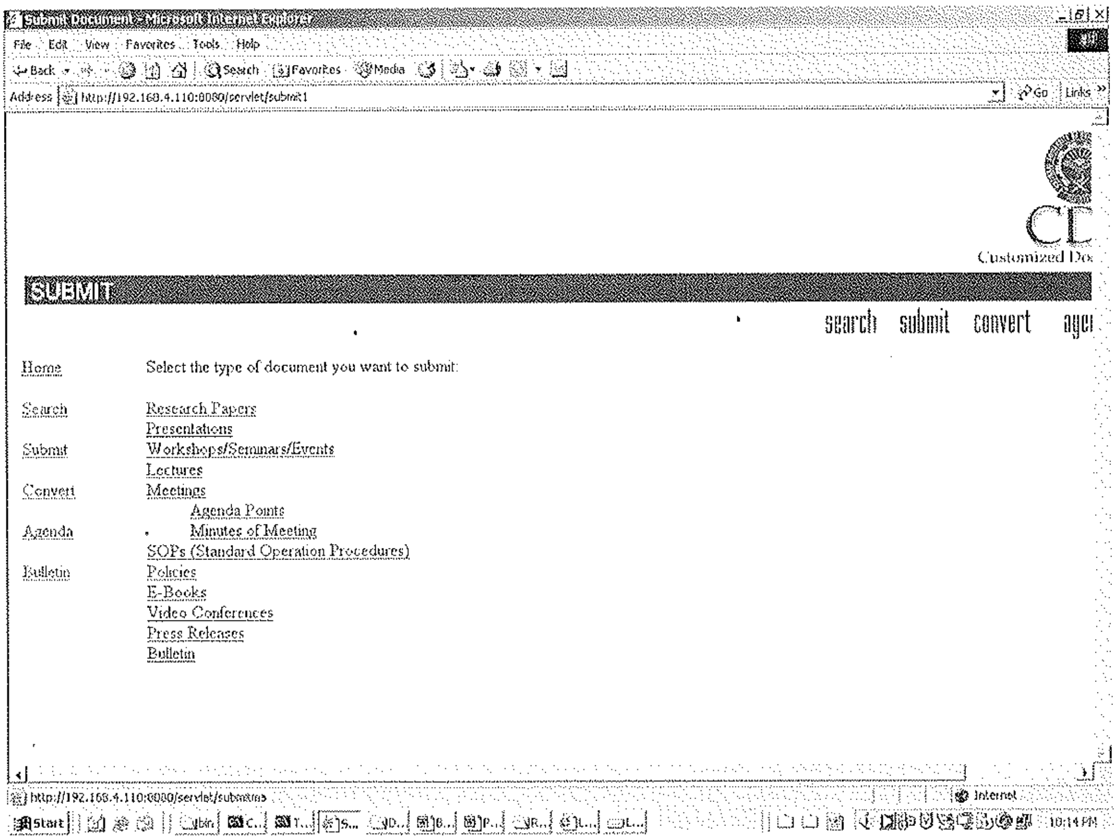
BIT-740 BIT 3-B NUST Institute Information Technology National University of Sciences and Technology Rawalpindi, **Pakistan** 2005 Project Committee Advisor. Wg
C:\Documents and Settings\Administrator\Desktop\Adnan1\Documentation\Project Proposal Document - Adnan Ali Khan.doc

NUST Institute of Information Technology

the requirement of the degree of Bachelors in Information Technology From NUST Institute of Information Technology National University of Sciences and Technology Rawalpindi, **Pakistan**
C:\Documents and Settings\Administrator\Desktop\Adnan1\Documentation\FYP Report.doc

Figure 5.6: System Validation (f)

To submit document, select the type of submission document:

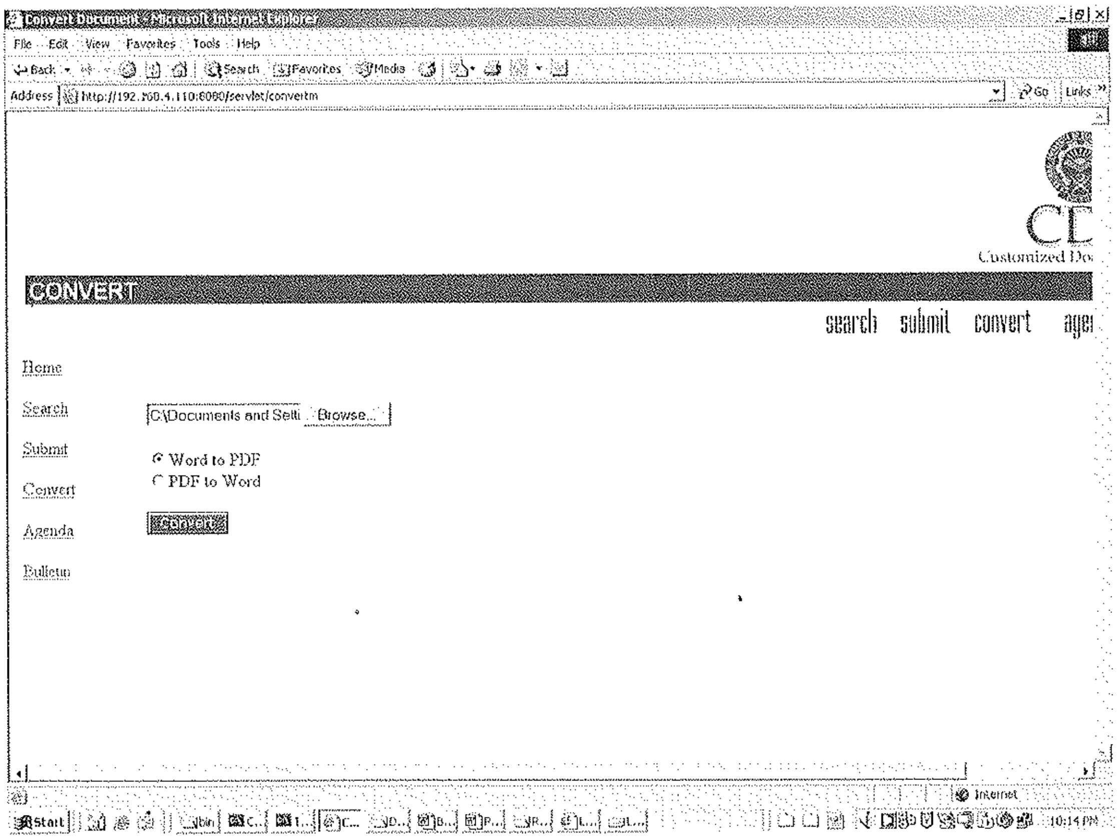

Submit Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://192.168.4.110:8080/servlet/submit1
Go Links
CD
Customized Do
SUBMIT
search submit convert agei
Home
Search
Submit
Convert
Agenda
Bulletin
http://192.168.4.110:8080/servlet/submitms
Internet
Start
10:14 PM

Select the type of document you want to submit:

Research Papers
Presentations
Workshops/Seminars/Events
Lectures
Meetings
    Agenda Points
    Minutes of Meeting
SOPs (Standard Operation Procedures)
Policies
E-Books
Video Conferences
Press Releases
Bulletin

Figure 5.7: System Validation (g)

For format conversion, select the file, and the format conversion type:

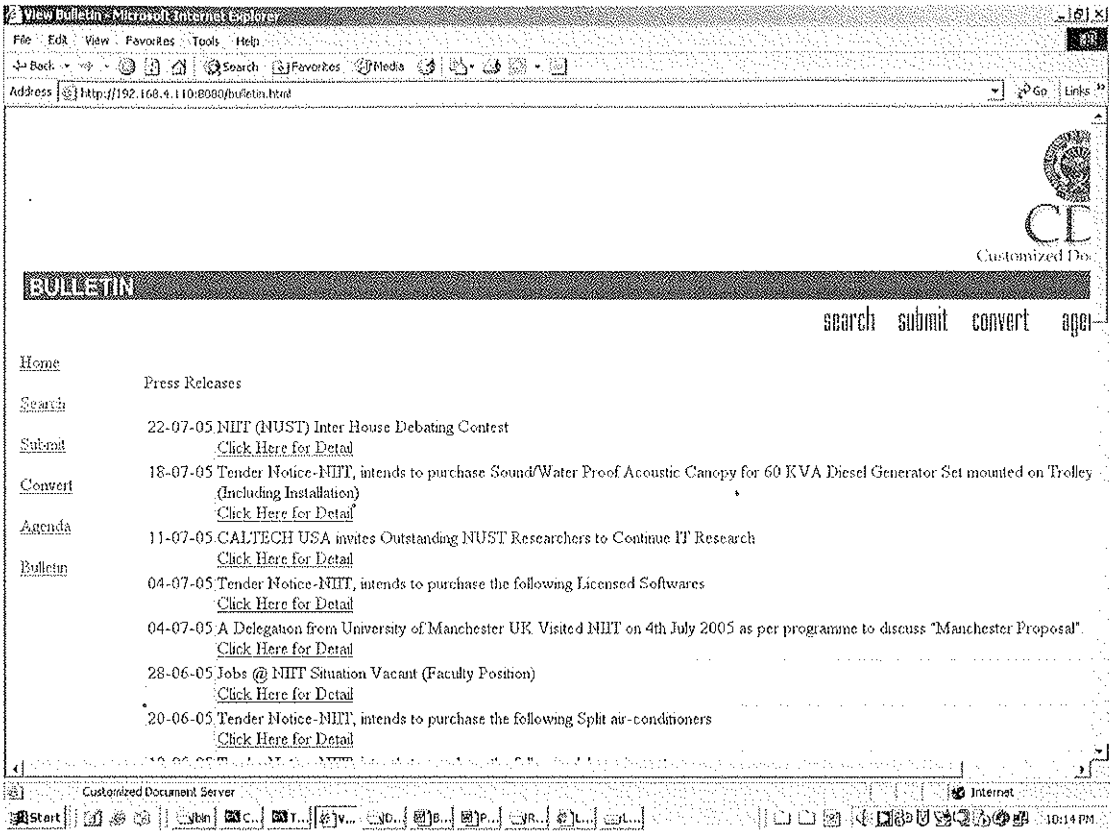

Convert Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://192.168.4.110:8080/servlet/convertm
Go Links
CD
Customized Do
CONVERT
search submit convert age
Home
Search
Submit
Convert
Agenda
Bulletin
Browse...
Convert
Internet
Start
10:14 PM

C:\Documents and Setti

● Word to PDF
○ PDF to Word

Figure 5.8: System Validation (h)

To view bulletin, click on 'Bulletin' button:


View Bulletin - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://192.168.4.110:8080/bulletin.html
Go Links
CD
Customized Do
BULLETIN
search submit convert age
Home
Search
Submit
Convert
Agenda
Bulletin
Customized Document Server
Internet
Start
10:14 PM

Press Releases

22-07-05 NIIT (NUST) Inter House Debating Contest
Click Here for Detail

18-07-05 Tender Notice-NIIT, intends to purchase Sound/Water Proof Acoustic Canopy for 60 KVA Diesel Generator Set mounted on Trolley (Including Installation)
Click Here for Detail

11-07-05 CALTECH USA invites Outstanding NUST Researchers to Continue IT Research
Click Here for Detail

04-07-05 Tender Notice-NIIT, intends to purchase the following Licensed Softwares
Click Here for Detail

04-07-05 A Delegation from University of Manchester UK Visited NIIT on 4th July 2005 as per programme to discuss "Manchester Proposal".
Click Here for Detail

28-06-05 Jobs @ NIIT Situation Vacant (Faculty Position)
Click Here for Detail

20-06-05 Tender Notice-NIIT, intends to purchase the following Split air-conditioners
Click Here for Detail

Figure 5.8: System Validation (i)

5.3 SUMMARY OF THE CHAPTER

This chapter provides an insight into how the project has been implemented. It also provides the validation that the project works, through different snapshots of the application.

Chapter 6

CONCLUSION AND RECOMMENDATION

The conclusion of different kind of analysis was presented in the previous chapter. So here we will summarize some of the ideas from previous chapters with the addition of few more. Recommendations regarding future work are also provided.

6.1 CONCLUSION

It is an excellent idea to make up a document server since it immensely helps students in research-oriented universities. Since enormous amount research work is being done at NIIT, a customized document server is a perfect remedy.

This project represents an efficient system to store documents, search them, and submit them. It has powerful search engine that helps to speedily search data.

The system is very easy to use even by people with little or no technological background (even though this is intended for people with technological knowledge).

It is an adaptable system, which can work in *Microsoft Windows* as well as in *Linux* (with little modifications) operating systems.

6.2 RECOMMENDATION

Though I have tried to make up a good and complete system, but there are always some new features that can be added to make the system even better. Though my system deals with session tracking, but maximum security is not provided by it. Since the Customized Document Server is supposed to be able to be accessible by almost every interested person, too much security constraints might have made it difficult for everyone to get access to the documents. But in the future, people might prefer applying more security constraints to the system.

6.3 SUMMARY OF THE CHAPTER

This chapter offers a conclusion to work I have done on the project. It also provides recommendations and suggestions for the future work on the related projects.

REFERENCES

[1] Jeffrey A. Hoffer, Mary B. Prescott and Fred R. McFadden, "Modern Database Management", Pearson Education Ltd., Edition 6, March 2002.

[2] Catherine Ricardo, "Database Systems: Principles, Design and Implementation", Maxwell MacMillan Ltd., August 2000.

[3] Shelley Greenhouse, "The Future of Database Indexing", American Society of Indexers, July 2000.